DOI - 10.32743/UniTech.2023.117.12.16418

ПРОГРЕСС В ОБРАБОТКЕ УЗБЕКСКОГО ЯЗЫКА: КОМПЛЕКСНЫЙ ПОДХОД К ИСПРАВЛЕНИЮ ГРАММАТИЧЕСКИХ ОШИБОК С ИСПОЛЬЗОВАНИЕМ МОДЕЛИ uzT5

Давронов Рифкат Рахимович
канд. техн. наук, ст. науч. сотр.,
Институт Математики им В.И. Романовского
АН Республики Узбекистан,
Республика Узбекистан, г. Ташкент
E-mail: rifqat.davronov@mathinst.uz

ADVANCING UZBEK LANGUAGE PROCESSING: A COMPREHENSIVE APPROACH TO GRAMMAR ERROR CORRECTION WITH THE uzT5 MODEL

Rifqat Davronov
Ph.D., Senior Researcher
V.I. Romanovsky Institute of Mathematics
of the Academy of Sciences of the Republic of Uzbekistan,
Republic of Uzbekistan, Tashkent

Я хотел бы выразить свою благодарность Фатиме Адиловой, Кушмуродову Самариддину, Сафарову Рузмату и Фуркату Низомову за их ценные обсуждения и помощь в обучении и оценке моделей.

АННОТАЦИЯ

В данной статье рассматривается проблема автоматизированного исправления грамматических ошибок (GEC) в узбекском языке, который мало изучен из-за своего статуса языка с недостаточными ресурсами. Несмотря на то, что узбекский язык занимает 54-е место в мире по числу говорящих, ему не хватает значительных обучающих данных для GEC, пробел, который эта исследовательская работа стремится заполнить. Мы представляем новый метод создания масштабной базы данных узбекского языка для обучения модели uzT5, рамочной структуры на основе T5, адаптированной для узбекского языка. Кроме того, мы представляем алгоритм, который преобразует правильные предложения в ошибочные для улучшения процесса обучения. Наш подход использует параллельные данные и техники NMT, что позволяет модели различать правильные и неправильные структуры предложений. Эта работа вносит значительный вклад в область, предлагая уникальный набор данных и методологию обучения с целью продвижения возможностей систем GEC в обработке текстов на узбекском языке.

ABSTRACT

This paper addresses the challenge of automated Grammar Error Correction (GEC) in the Uzbek language, which has been understudied due to its status as a low-resource language. Despite being the 54th most spoken language globally, Uzbek lacks substantial training data for GEC, a gap that this research aims to fill. We introduce a novel method for generating a large-scale Uzbek language database to train the uzT5 model, a T5-based framework adapted for Uzbek. Additionally, we present an algorithm that converts correct sentences into erroneous ones to enhance the training process. Our approach utilizes parallel data and NMT techniques, enabling the model to differentiate between correct and incorrect sentence structures. This work significantly contributes to the field by proposing a unique dataset and training methodology, aiming to advance the capabilities of GEC systems in processing Uzbek language texts.

Ключевые слова: многоязычность, определение языка, GEC, T5, Uzbek language texts.
Keywords: multilingual, language detection, Grammar Error Correction, T5, Uzbek language texts.

1. Введение

Грамматическая структура является критически важным компонентом человеческого общения, необходимым для обеспечения ясности и понятности письменной и устной речи [14]. Ошибки в грамматическом построении могут привести к недопониманию и затруднениям в восприятии у аудитории. Автоматизированное исправление грамматических неточностей стало ключевой темой многих исследований в области обработки естественного языка (NLP). В последнее время языки, такие как английский и китайский, привлекли значительное внимание

Библиографическое описание: Давронов Р.Р. ПРОГРЕСС В ОБРАБОТКЕ УЗБЕКСКОГО ЯЗЫКА: КОМПЛЕКСНЫЙ ПОДХОД К ИСПРАВЛЕНИЮ ГРАММАТИЧЕСКИХ ОШИБОК С ИСПОЛЬЗОВАНИЕМ МОДЕЛИ uzT5 // Universum: технические науки : электрон. научн. журн. 2023. 12(117). URL: *https://7universum.com/ru/tech/archive/item/16418*

в исследованиях благодаря появлению обширных предварительно обученных моделей и значительных наборов данных, нацеленных на исправление грамматических ошибок (GEC), достигая уровней производительности, сопоставимых с человеческой компетенцией [7; 8]. В отличие от этого, языки, такие как арабский, русский и узбекский, не были так широко исследованы, что в основном объясняется их классификацией как языков с недостаточными ресурсами и ограниченными обучающими данными. Узбекский язык, занимающий 54-е место в мире по распространенности [18] и насчитывающий более 44 миллионов говорящих [15] испытывает ограничения в исследованиях, особенно в области GEC, из-за этого дефицита данных.

Недавние исследования в области автоматического исправления грамматических ошибок (GEC) на основе трансформерных моделей привели к значительным достижениям и прозрениям.

Другое исследование сочетало модель Transformer с Генеративной Адверсариальной Сетью для автоматического английского GEC. Этот метод был проверен на наборах данных и способствовал дальнейшему улучшению GEC, подчеркивая потенциал сочетания трансформеров с другими методами глубокого обучения [11]. Недавние работы подчеркнули важность языкового моделирования в GEC, указывая на то, что сравнение вероятностей предложенных правок может привести к хорошей производительности. Этот подход использует возможности понимания языка трансформерных моделей для оценки и выбора наиболее вероятных исправлений [6].

Использование трансформеров в GEC не ограничивается английским языком. Например, недавние подходы в корейском и индонезийском GEC адаптировали трансформеры, демонстрируя конкурентоспособную и перспективную производительность по сравнению с традиционными RNN-основанными моделями кодировщика-декодера. Эти достижения подчеркивают гибкость и эффективность модели на разных языках [9; 12].

Несмотря на эти достижения, остаются проблемы. Одно исследование показало, что стандартная модель GEC на основе Transformer не смогла реализовать грамматическую обобщенность даже в простых условиях с ограниченным словарным запасом и синтаксисом. Это указывает на ограничения способности модели обобщать и применять грамматические правила за пределами предоставленных обучающих примеров [10]. GEC на узбекском языке все еще находится в стадии развития, и проведено лишь несколько исследований, таких как исследования Tahrirchi [16]. Они использовали GEGToR [17], основанный на трансформере, соединенном с двумя линейными выходными слоями, ответственными за обнаружение ошибок и исправление ошибок, включающими кодировщик.

Методологии GEC, использующие подходы нейронного машинного перевода (NMT), используют параллельные наборы данных для облегчения процесса обучения, позволяя системе ассоциировать входные данные источника (правильные предложения) с их соответствующими целями (неправильные предложения), как показано на рисунке 1.

Он **пойдет** на улицу без зонта, несмотря на то что **идет** дождь.
Система GEC
Он пойдёт на улицу без зонта, несмотря на то, что идёт дождь.

Рисунок 1. Иллюстрация системы для исправления грамматических ошибок

2. Методы

В этой статье мы подробно описываем наши методы предобработки, применяемые к лингвистическим корпусам, полученным из различных источников и подвергнутым различной обработке. Перед выбором подмножеств для конкретных задач эти корпуса проходят тщательную фазу предобработки. Эта фаза начинается с сегментации каждого документа на предложения с использованием нашего Алгоритма-1.

Для снижения шума мы сначала удаляем элементы, способствующие текстовому шуму. Это включает в себя извлечение несоответствующего узбекского текста, хэштегов (например, #JMT), HTML-элементов (например, тегов <br>), символов, нарушающих поток текста (например, U+200B и

U+FEFF), и избыточных пробелов. Кроме того, предложения, содержащие избыток иностранных символов (свыше порога в 10), отфильтровываются. В частности, в корпусах, полученных из журналистских материалов, мы также удаляем метаданные, такие как кредиты журналистов, описания изображений и примечания об авторском праве.

Для исключения вредного контента из модели используются автоматизированные инструменты для удаления предложений с потенциально вредной или предвзятой информацией.

Кроме того, в связи с проблемами конфиденциальности, мы активно удаляем любое предложение, которое может содержать личные данные. Обнаружение таких предложений осуществляется с помощью регулярных выражений, определяющих адреса электронной почты, URL-адреса и ссылки на пользователей (например, "@xxxx").

Алгоритма-1.

функция разделить_на_предложения(текст)

Определить регулярные выражения для алфавитов, префиксов, суффиксов, начальных слов, аббревиатур, веб-сайтов и цифр.

Предобработка текста:

1. Добавить пробелы перед текстом и после него.
2. Заменить символы новой строки на пробелы.
3. Заменить определенные шаблоны (префиксы, веб-сайты, шаблоны чисел и т.д.) на заполнитель "<prd>", чтобы сохранить границы предложений.

Обработка особых случаев:

1. Заменить вхождения многоточий и определенных аббревиатур (например, "Ph.D", "кв.м", "кв.км") на заполнители.
2. Отрегулировать пунктуацию возле кавычек и других символов.

Заменить завершающие символы предложений (точки, восклицательные и вопросительные знаки) на "<stop>":

1. Преобразовать различные знаки препинания, такие как "!!!", "!?", "?!", и т.д., в уникальные заполнители.
2. Заменить эти заполнители на "<stop>".

Восстановить точки из заполнителей "<prd>".

Разделить текст на предложения:

1. Разделить текст в каждом месте "<stop>".
2. Удалить последующее пустое предложение, если оно присутствует.
3. Удалить ведущие и завершающие пробелы из каждого предложения.

Вернуть список предложений.

конец функции

Модель. В нашем исследовании мы разрабатываем uzT5, основанную на фундаментальной архитектуре T5 от Раффела и др. (2019). Эта модель, использующая рамочную структуру кодировщика-декодера Transformer, признана за её исключительную производительность в различных задачах генерации естественного языка (NLG). Хотя uzT5 доступна в нескольких конфигурациях, наше исследование конкретно использует базовую модель, которая включает 223 миллиона параметров.

Предварительное обучение T5. Предварительная фаза обучения модели uzT5 включала корпус C4 и дополнительные данные, полученные из интернета, в основном из Common Crawl, общим объемом около 3 ГБ. Для всестороннего понимания набора данных mC4 рекомендуется знаковая работа Сюэ и др. (2020). Основной акцент этой фазы предварительного обучения сосредоточен на упражнении по предсказанию слов, варианте цели маскированного языка, адаптированном специально для авторегрессивных последовательно-последовательных рамок. Ниже приведен пример этого упражнения по предсказанию слов.

Пример: Входные данные: Комплексный подход [x] Исправление с помощью [y] Цель: [x] к Исправлению Грамматических Ошибок [y] Модель uzT5. Полная модель uzT5 была обучена в течение 1,8 миллиона итераций с использованием размеров пакетов по 512 входных последовательностей, каждая из которых имела максимальный предел в 512 токенов. Это обучение включало обработку примерно 32 000 токенов в общем количестве. Для всех экспериментальных усилий использовалась широко доступная контрольная точка T5-base.

Техника создания синтетических данных. Одной из основных проблем в GEC, особенно для языков с недостаточными ресурсами, является недостаточность обучающих наборов данных. Чтобы противостоять этой проблеме, был разработан полуавтоматический метод создания искусственных данных, тем самым увеличивая объем обучающих данных. Эти искусственные данные использовались в предварительном обучении модели GEC, происходящей из открыто доступного набора данных узбекского новостного сайта, состоящего из 712000 предложений и общим объемом 512 МБ.

Первым шагом было объединение данных в один файл и их стандартизация путем исправления не-UTF8 символов, упоминаний, хэштегов, гиперссылок и избыточных пробелов, при этом сохраняя восклицания, цифры и другие знаки препинания. Данные в целевом файле были отделены предложениями, состоящими из 10 до 40 слов. Затем были созданы три различных синтетических набора данных на основе различных уровней нормализации данных. В первом наборе данных нормализация проводилась как в исходных, так и в целевых предложениях. Во втором наборе данных нормализация проводилась только с целевой стороны, а в третьем наборе данные сохранялись в их первоначальном состоянии.

Методология включает в себя три различные тактики. Первая включает введение орфографических ошибок путем добавления или удаления случайных символов в словах. Вторая стратегия предусматривает создание ошибок пунктуации путем преобразования предложений в теги частей речи (POS), за которым следует добавление или опущение пунктуации из определенного списка. Третья тактика случайным образом меняет пары слов в предложениях для введения семантических неточностей. Наконец, эти искусственно созданные предложения были сопоставлены с их оригинальными аналогами для формирования синтетических обучающих наборов GEC, демонстрирующих разнообразие сценариев обучения.

Детали этого полуавтоматического подхода изложены в Алгоритме 2.

Алгоритм 2.

```
Функция ДобавитьСимвол(W_i):
// Добавить случайный символ в строку W_i
Выбрать случайный символ c_i из W_i
Выбрать случайный символ c_i_prime из букв ASCII
Найти индекс c_i в W_i и увеличить его на 1
Вставить c_i_prime в W_i на новый индекс
Вернуть измененную строку W_i_prime

Функция УдалитьСимвол(W_i):
// Удалить случайный символ из строки W_i
Если длина W_i больше 1:
Выбрать случайный символ c_i из W_i
Удалить первое вхождение c_i из W_i
Вернуть измененную строку W_i_prime
Иначе:
Вернуть W_i как есть

Функция ДобавитьПунктуацию(Y_i):
// Вставить случайный знак пунктуации в строку Y_i
Определить список знаков пунктуации Plst
Выбрать случайный знак пунктуации Plst_i из Plst
Выбрать случайный индекс в Y_i
Вставить Plst_i в Y_i на выбранный индекс
Вернуть измененную строку Y_i_prime

Функция ПоменятьСлова(Y_i):
// Поменять местами два случайных слова в строке Y_i
Разделить Y_i на список слов
Если в списке больше одного слова:
Случайным образом выбрать два разных индекса idx1
и idx2
Поменять местами слова на idx1 и idx2
Объединить слова обратно в строку Y_i_prime
Вернуть Y_i_prime
Иначе:
Вернуть Y_i как есть

Функция ПроцессЗамешательства(X, alpha=0.1):
// Применить процесс замешательства к строке X
Установить Y_prime равным X
Вычислить N как alpha, умноженное на количество
слов в X
Определить список функций fns, содержащий Добавить
Символ и УдалитьСимвол
Повторять N раз:
Выбрать случайное слово W_i из Y_prime
Выбрать случайную функцию fn из fns
Заменить W_i в Y_prime на результат fn(W_i)
Применить функцию ДобавитьПунктуацию к Y_prime
Применить функцию ПоменятьСлова к Y_prime
Установить X равным Y_prime
Вернуть X
```

Модель GEC. Данные для дополнительного обучения. Для нашего исследования мы использовали корпус, состоящий из 512 МБ данных с сайтов kun.uz и daryo.uz. После увеличения данных общий объем синтетических параллельных данных составил 1,56 ГБ. Итоговый обучающий набор данных был разделен, из которых 6 043 864 примера были выделены для обучения и 671 541 пример – для развивающего набора.

Обучение. В нашем исследовании мы изучали различные методы обучения, все они основаны на рамочной модели дополнительного обучения T5. В фазе дополнительного обучения GEC мы поддерживали постоянную скорость обучения 0,001. Предварительная фаза обучения продолжалась до достижения сходимости, в то время как фаза дополнительного обучения была продлена до заметного снижения точности точного совпадения на развивающем наборе, что обычно происходило после 1 888 710 шагов или завершения 10 эпох.

3. Результат

Мы использовали набор данных, содержащий ошибки узбекской грамматики, который мы разработали внутри компании. Этот набор данных разделен таким образом, что 90% выделено для обучения, в то время как оставшиеся 10% используются в качестве развивающего набора. Результаты, полученные из этого набора данных, представлены в Таблице 1.

Таблица 1.

Модель GEC (rouge1, rouge2, rougeL, rougeLsum)

loss	rouge1	rouge2	rougeL	rougeLsum
0,018	72.8143	70.5835	72.5677	72.5675

В процессе разработки и оценки нашей модели NLP мы отслеживали её производительность во времени, используя метрики ROUGE (Recall-Oriented Understudy for Gisting Evaluation), которые критически важны для оценки способности модели понимать и генерировать текст. Графики показывают прогресс трех различных метрик ROUGE: ROUGE-1, ROUGE-2 и ROUGE-L, которые соответственно измеряют перекрытие униграмм, биграмм и наиболее длинной общей подпоследовательности между выводом модели и эталонным текстом. Эти измерения проводятся в различных контрольных точках, называемых 'глобальными шагами', во время обучения модели.

Каждый график показывает постоянное увеличение показателя метрики, что свидетельствует о непрерывном улучшении производительности модели с точки зрения точности и полноты последовательностей слов различной длины. Обучение модели

проводилось в диапазоне от 500 тыс. до 1,5 млн глобальных шагов, при этом все метрики показывали заметный восходящий тренд, указывающий на повышение профессионализма модели в захвате нюансов языка, на котором она обучается. Результаты этих оценок являются ключевыми для определения готовности модели к внедрению в реальные приложения и предоставляют представления о динамике обучения на протяжении обширного количества итераций.

До нашего исследования задача исправления грамматических ошибок для узбекского языка не была решена. Следовательно, не было возможности сравнить нашу модель исправления грамматических ошибок (GEC) с другими моделями.

4. Заключение

В настоящее время не существует специализированной базы данных грамматических ошибок на узбекском языке. Чтобы решить эту проблему, мы сотрудничали с лингвистическими экспертами для выявления и идентификации грамматических ошибок в узбекском языке. Этот процесс включал не только синтетические ошибки, но и те, которые наблюдались на платформах социальных медиа. Каждый пример в нашем наборе данных содержит только одну ошибку, что позволяет модели обнаруживать единственную ошибку в предложении. Сейчас мы разрабатываем новую базу данных Исправления Грамматических Ошибок (GEC), где каждый пример обогащается по крайней мере тремя типами ошибок. Результаты этой разработки будут включены в последующую статью.

Список литературы:

1. Adilova F.T., Davronov R.R., Safarov R.A. UzRoberta: An Uzbek Language Pre-Trained Model // Universum: технические науки : электрон. научн. журн. 2023. Vol. 10 (115). URL: https://7universum.com/ru/tech/archive/item/16028
2. Davronov R., Adilova F. BERT-Based Drug Structure Presentation: A Comparison of Tokenizers // AIP Conference. 2023. Vol. 2781.
3. Davronov R.R., Safarov R.A., Abdumalikov S.Q. Application of transformer model architecture in the new drugs design // International Conference on Information Science and Communications Technologies (ICISCT). 2021.
4. Davronov R.R., Safarov R.A., Abdumalikov S.Q. Predicting the activity and properties of chemicals based on RoBERTa // International Conference on Information Science and Communications Technologies (ICISCT). 2021. Pp. 1−4.
5. Davronov, R., Adilova, F. A comparative analysis of the ensemble methods for drug design // AIP Conference. 2021 Vol. 2365.
6. Dimitris Alikaniotis, Vipul Raheja. The Unreasonable Effectiveness of Transformer Language Models in Grammatical Error Correction // Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications. 2019. Florence, Italy. Pp. 127–133. http://dx.doi.org/10.18653/v1/W19-4412
7. Grundkiewicz R., Junczys-Dowmunt M. Near Human-Level Performance in Grammatical Error Correction with Hybrid Machine Translation // Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. New Orleans, LA, USA. 1−6 June 2018. Vol. 2. Pp. 284—290. https://doi.org/10.18653/vl/N18-2046.
8. Kiyono S., Suzuki, J., Mizumoto, T., Inui, K. Massive exploration of pseudo data for grammatical error correction // IEEE/ACM Transactions on Audio, Speech, and Language Processing. 2020. Vol. 28. Pp. 2134−2145.
9. Lee M, Shin H, Lee D, Choi S-P. Korean Grammatical Error Correction Based on Transformer with Copying Mechanisms and Grammatical Noise Implantation Methods // Sensors. 2021. Vol. 21(8). Pp. 26−58. https://doi.org/10.3390/s21082658
10. Masato Mita, Hitomi Yanaka, Do Grammatical Error Correction Models Realize Grammatical Generalization? [Электронный ресурс]. – Режим доступа: https://doi.org/10.48550/arXiv.2106.03031 (дата обращения: 28.11.2023).
11. Mengyang Qin. A study on automatic correction of English grammar errors based on deep learning [Электронный ресурс]. – Режим доступа: https://doi.org/10.1515/jisys-2022-0052
12. Musyafa, A., Gao Y., Solyman A., Wu C., Khan S. Automatic Correction of Indonesian Grammatical Errors Based on Transformer // Applied Sciences. 2022. Vol. 12. 10380. https://doi.org/10.3390/app122010380
13. Satoru Katsumata, Mamoru Komachi, Stronger Baselines for Grammatical Error Correction Using a Pretrained Encoder-Decoder Model [Электронный ресурс]. – Режим доступа: https://aclanthology.org/2020.aacl-main.83/ (дата обращения: 27.11.2023).
14. Solyman A., Wang Z., Tao, Q. Proposed Model for Arabic Grammar Error Correction Based on Convolutional Neural Network // International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE). Khartoum North, Sudan. 21-23 September 2019. Pp. 1−6.

15. [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Uzbek_language (дата обращения: 29.11.2023).
16. [Электронный ресурс]. – Режим доступа: https://tahrirchi.uz/grammatika-tekshiruvi (дата обращения: 30.11.2023).
17. [Электронный ресурс]. – Режим доступа: https://www.grammarly.com/blog/engineering/experimenting-with-gector (дата обращения: 23.11.2023).
18. [Электронный ресурс]. – Режим доступа: https://www.visualcapitalist.com/100-most-spoken-languages/ (дата обращения: 28.11.2023).